



The longer and more intensively you work with your data, the more codes are likely to be generated and the more coding is carried out. This chapter deals with different types of category systems and different ways of arriving at a coding frame that is optimally suited for analysis. The two opposing poles of category formation are usually referred to as deductive and inductive category formation. In the former case, the categories are concept-based, i.e., defined before the actual analysis of the empirical data starts. In the second case, the categories are developed based on the empirical data. Inductive, data-based category building is very effectively supported in MAXQDA by the “Creative Coding” function. When working with categories, the code definitions play a very important role; they are used to record what a code means and when exactly it is assigned. Because categories play such a central role in many analysis methods, one should take sufficient time over the construction of the category system.

In This Chapter

- Learning about different types of coding frames
- Organizing hierarchical coding frames
- Getting to know the procedure for deductive category formation
- Developing categories based on data, forming inductive categories
- Creating code definitions and a codebook
- Working with the “Creative Coding” function
- Rethinking and structuring your coding frame

Different Types of Coding Frames

Codes can have very different forms, as described in Chap. 6; sometimes they consist of a single word or abbreviation and sometimes of several words or even a complete set of statements. Codes can be simple labels or names for complex constructs. Codes can be very concrete and very general or exhibit varying degrees of abstraction. Depending on the nature of the codes and their various functions, working with codes can also be very different. When coding, especially when working with the open coding technique, the number of codes can swiftly become confusing, which raises the question of how to organize them. The entirety of all these codes is also referred to as a “coding frame,” “category system,” or “code system.” The code system can be designed in three different ways: as a linear list, as a hierarchical structure, or as a network.

A linear list is the simplest structure; here all codes are on one level and are lined up in a list, like this:

- Environmental attitudes
- Environmental knowledge
- Environmental behavior
- Level of personal concern
- Personal CO₂ balance
- Membership in an environmental protection organization
- Greenpeace member
- Energy saving
- Mobility behavior
- Consumption patterns
- Avoiding packaging
- Knowledge about nature, animals, and plants

Such *linear lists* quickly become unmanageable and offer few possibilities for creating a structure. To create an order beyond merely sorting the codes alphabetically, you can often make things easier by defining appropriate word combinations for subordinate codes, such as “Environmental behavior, energy saving,” and “Environmental behavior, mobility,” or by working with abbreviations for the parent code, e.g., “Enb_energy saving,” “Enb_mobility,” etc. Designing a code system like this is better than a simple linear list, but such a structure is still very limited in its possibilities.

A *hierarchical code system*, as supported by MAXQDA, is much more flexible. This consists of top-level codes and multiple levels of subcategories. In general, (almost) any number of levels can be provided for such code systems, but in practice two to four levels are usually sufficient. Figure 8.1 shows such a hierarchical code system in the typical MAXQDA representation. In this code system, two main categories (top-level codes) are defined, namely, “Biggest world problems” and “Social influence.” Both codes have subcodes: the first main category “Biggest world problems” has the subcodes “Climate,” “Resources: scarcity, distribution,

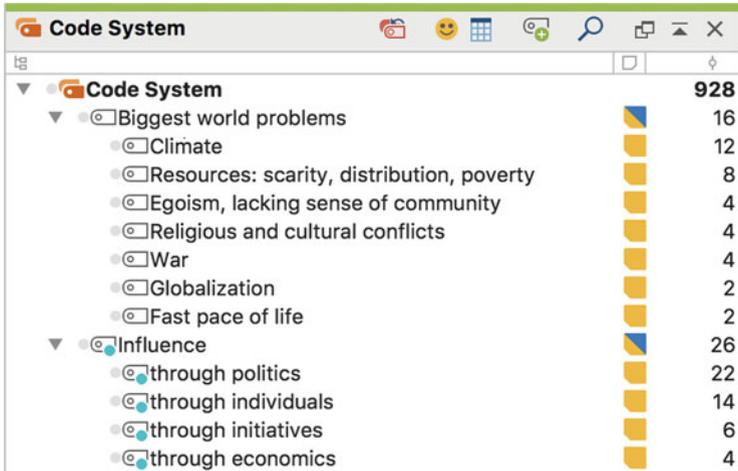


Fig. 8.1 Hierarchical category system with two levels in MAXQDA

poverty,” and five more. A memo symbol is displayed to the right of each code. A double-click on this symbol opens the code memo, i.e., a detailed description of the respective code.

With such hierarchical category systems, you can gain a much better overview, for example, by hiding the sublevels. However, it is also important to note that the hierarchical structure is particularly useful for search processes and complex queries. For example, MAXQDA can compile all the text passages in which there is an overlap of certain top-level codes. Here, all the conceivable overlaps of subcodes can be examined, for instance, the simultaneous occurrence of “Biggest world problems > Climate” and “Influence > through economics.” In a linear list, searching for subcodes that overlap with the subcodes of another top-level code would involve immense effort.

The structure as a *code network* is the third option for structuring a category system. In general, a network structure consists of a set of elements (nodes) that are linked to each other via connections (edges). The difference to a hierarchical structure is that there are no restrictions on these connections, whereas in a hierarchical structure, a subcode cannot be a subcode of several other parent codes at the same time.

MAXQDA can also be used to create category systems with a network structure, but not in the “Code System” window, where you can only set up and manage linear lists and hierarchical code systems. Network structures can be created with the help of the visualization tool MAXMaps. In the coding phase, you can code directly with this network (see Chap. 17).

Paths to a Structured Coding Frame

When working with MAXQDA, you will inevitably ask yourself the crucial question: “So, what about these categories, then?” More specifically, you may ask, “How do I even come up with my categories?”, “How many categories do I need for my analysis?” or “Which steps do I need to take to form my categories?”

How exactly to proceed when forming your categories depends first and foremost on the research question, the objective of your research, and the previous knowledge you have about the subject area of your research. The stronger the orientation to a theory, the more extensive the previous knowledge; the more specific the questions and the more precise any potentially already existing hypotheses, the better it is to form categories before the analysis of the collected data. This type of category formation is also called *concept-driven category formation* or *deductive category formation*. The decisive factor here is that a meaningful structure for the content of the data has already been established before it is coded. This can be a theory or a hypothesis but also an interview guide or an existing structure in the given field of research. Sometimes deductive category formation is ascribed the attribute of “theory-oriented.” However, this is not necessarily the case: concept-driven category building can be theory-oriented but does not have to be.

The counterpart to this approach is the *data-driven* or *inductive category formation*, in which the categories are formed directly on the basis of the data. The category system usually emerges as a hierarchical system in an iterative process that runs through several cycles. In methodological literature, the pairing of the terms inductive versus deductive category formation has become established in general usage, although the use of these terms which originate from philosophy to describe the practical procedure of category formation is by no means optimal. Like Schreier (2012, pp. 84–87), we prefer the more appropriate terms *concept-driven* for deductive and *data-driven* for inductive strategies of category formation (Schreier, 2012, pp. 84–87).

Creating Concept-Driven Categories (Deductive Category Formation)

How does the concept-driven formation of categories work? As described above, the starting point is an extensive structuring of the research questions that already exist before the coding phase. Frequently, such structuring has already determined the interview guidelines used to collect the data. A simple example is that before MAXQDA workshops we always send the participants an e-mail and ask them to answer some questions like the following:

1. Why are you attending the workshop? What are your goals, what do you want to learn?
2. Have you already gained experience with MAXQDA (or any other QDA software)? Please briefly describe your level of experience.

3. What kind of qualitative data do you want to analyze?
4. Here you can list up to five more questions that you would like to discuss during the workshop:

The following categories can easily be derived from these four questions:

- Goal of participation
- Experience with MAXQDA
- Type of data to be analyzed
- Questions to discuss in the workshop

The first step of the analysis is to code the corresponding text passages of the answers of the participants with these four codes. This first deductive step is followed by a second inductive step. For example, the “questions for the workshop” mentioned by the participants will be grouped and, in view of the design of the planned workshop, combined into question blocks. These blocks are then created in MAXQDA’s “Code System” as subcodes of the top-level code “Questions to discuss in the workshop.”

In a comprehensive guide, the definition of categories may be more complicated, but the principle is the same, namely, to derive categories for the analysis from the questions in the guide, which have already determined the structure of the data collected. It is very helpful to keep your own research questions in mind and to construct the categories in such a way that a suitable structure for the research report can be developed later. Of course, the analysis of data in qualitative research should have an open character, but foresight in planning is always useful.

Another option for concept-driven category formation—as an alternative to deriving the categories from the interview guide—is the formation of categories based on a specific theory or the current state of research. The former is rather rare in practice, but there are both good examples and good arguments in favor of this approach (C. Hopf, 2016). If the aim of research is to examine a theory or, as in the case of Hopf, the question of whether a certain theory (here the theory of attachment) can explain phenomena (here right-wing radical thinking), then it is only logical to derive the analysis categories from this theory.

It may also make sense to make the current state of research the basis of category formation. Kuckartz (2016, pp. 67–72) reports on the development of a coding frame on the topic of quality of life, in which the following categories were deductively formed in a rather elaborate group procedure based on the state of research:

- Work and occupation (also includes labor market and career opportunities)
- Education (also includes educational opportunities)
- Political freedom in the sense of participation and political participation
- Health
- Individual freedom in the sense of self-determination, self-realization, and free choice of lifestyle
- Culture (also includes cultural and leisure activities)

- Standard of living and wealth
- Security (in regard to war, civil war, crime, personal assaults, but also from personal poverty)
- Social inclusion
- Environment, nature, sustainability
- Work-life balance, time prosperity

These 11 categories cover all those areas that are relevant to quality of life according to the current state of research. The process of developing this category system made it clear to our group of researchers that the claim of consistency and reliability cannot be made when designing a category system. In contrast to what the term may suggest, such deductive categorization is also a constructive process in which researchers act on the basis of their previous knowledge and their specific views. An attempt at reaching a consensus here would be just as misguided as an attempt to have different working groups within a quantitative research framework constructs the same questionnaire with the same questions. However it is done, the category system must be formed in such a way that the categories are clearly defined and that the coders can use the categories in a reliable manner. The category definitions play a key role here. These are used to determine as precisely as possible when a particular category is to be applied.

Category Definitions and Code Memos

Working with categories is central to many projects; it requires a lot of brain work, a lot of time, and careful work. This is true for different methods and research styles but especially for projects that work according to the grounded theory approach or the method of qualitative content analysis. In both cases, it is important to describe the meaning of a category as clearly as possible, for example, by writing an appropriate memo. The grounded theory calls this a *code memo* (Charmaz, 2006, pp. 72–85; Glaser & Strauss, 2009), while in qualitative content analysis, *category definition* is the preferred term. At the beginning of a grounded theory analysis, a memo may only contain a few words or keywords and ideas. As the analysis progresses, code memos become more complex and sometimes take the form of highly differentiated theoretical thoughts on a specific key area of content.

The approach in grounded theory differs significantly from other methods such as qualitative content analysis. The latter is a more rule-based method in which it is particularly important to formulate category definitions as precisely as possible in the process of constructing the category system. Regardless of how the categories were developed, whether inductively based on the material or in advance without empirical data, each category should be defined precisely in a qualitative content analysis. The general structure displayed in Fig. 8.2 is recommended for a category definition.

Category definitions have a dual function: firstly, they document the framework of the analysis for the scientific community (and also for the reviewers of a publication), and, secondly, they form the basis of the coding guide used by the coders. This means that the better the definitions and the clearer the examples, the better the

Name of the category:	As concise a name as possible
Description of the category:	Description of the category, possibly with a theoretical linkage
Application of the category:	“Category x” is coded if the following aspects are present ...
Examples of applications:	Quotations with reference (document, paragraph)
Further applications (optional):	The category is also assigned if ... Quotations with reference (document, paragraph)
Differentiation from other categories (optional):	The category is not coded if... in this case, “Category y” is used Quotations with reference (document, paragraph)

Fig. 8.2 General scheme for category definitions

coding and the higher the probability of achieving a good match between the coders will be.

MAXQDA also offers the option of compiling all categories and their definitions in a category manual, a so-called codebook, via the function *Reports > Codebook*. This is particularly important for masters’ theses and dissertations, as it is an excellent way of documenting the rigor and accuracy with which work has been carried out. Figure 8.3 shows a code memo written in MAXQDA containing the name of the code, author and creation date, the code definition, and a “perfect example.” This is the term used to describe text passages that are prototypical for the assignment of the code in question.

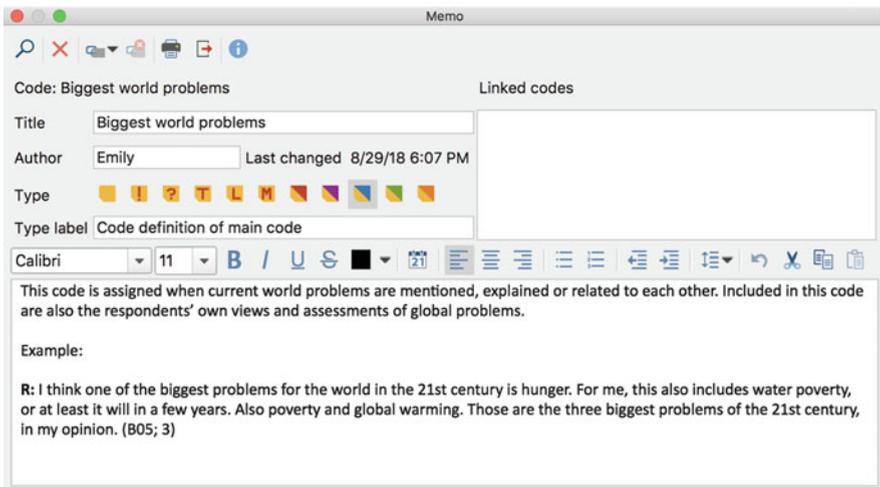


Fig. 8.3 Defining a code in a code memo

Creating Data-Driven Categories (Inductive Category Formation)

Typical for qualitative research is the formation of categories based directly on the data, which is called data-based or inductive category formation. Inductive does not mean, however, that the categories simply flow to you from the data, as it were, but it is an active process that is inconceivable without the active engagement, knowledge and understanding of the material, and linguistic competence of those involved in the category formation. Since the formation of categories thus depends on individual category-building competence and active engagement, it is difficult to postulate intersubjective agreement in the construction of a category system. If several people, alone or in groups, form categories based on the same data, the categories will be partly similar, partly even the same, but also partly different. Hence, any potential demand for inductive coding by several people or members of a team to result in the formation of the same categories cannot be met. It therefore also does not make sense to calculate coefficients of agreement or the intercoder reliability in general for purposes of category *formation*. Nevertheless, this does not mean that several people should not be involved in the categorization process. On the contrary, it is highly recommended that several people be involved in the process of category development. They should first develop category proposals independently of each other and then exchange ideas in order to fully exploit the group's creative potential.

The Process of Data-Driven Category Building

There are a number of approaches for data-based category building (z.B. Charmaz, 2006; Kuckartz, 2014; Mayring, 2014; Strauss & Corbin, 1990). A guideline for how to proceed in the research process can be found in Kuckartz (2014, pp. 58–60), where six phases of data-driven category formation are described for text data:

1. Determine the objective of category building on the basis of the research question.
2. Determine the type of categories and the level of abstraction.
3. Familiarize yourself with the data, and determine the type of coding unit, i.e., the scope of the material to be coded.
4. Process the texts sequentially, and create categories while working with a text, assigning either new categories or existing categories.
5. Group the formed codes, systematize and organize the category system, and make sure that the categories form a meaningful whole.
6. Set (fix) the category system.

At the outset of the data-driven category formation process, the “Code System” will still be empty, i.e., new codes must be generated. As the category-building process progresses, you will often combine these codes into more abstract codes or define new top-level codes and assign already generated codes to them.

Create a New Code While Working with a Text

If you want to create new codes during inductive coding of the data, the fastest way to do this is to press **Alt+W** (Windows) or **⌘ + ⌘ + W** (Mac). Chapter 6 describes several alternative ways of creating new codes in addition to this key combination.

Assigning a Text Passage to an Existing Code

If codes that have already been defined in the “Code System” are to be assigned, this can be done quite conveniently by dragging and dropping the selected text onto the respective code or vice versa, the code onto the selected segment.

Merge Codes

If two codes have very similar meanings, which is not uncommon in open coding following the grounded theory method, it makes sense to merge the codes. Usually the code name will be adapted accordingly and replaced by a more general term. With inductive coding, there is automatically at least one coded text passage for each code; “empty codes” without assigned text passages do not exist here. Merging codes always takes place in the following three steps:

1. Right-click on the code to be merged (“Code A”), and select the option *Move Coded Segments from the context menu*.
2. Click on the code you want to merge Code A with, and select the option *Move Coded Segments from “Code A” from the context menu*. Now the coded segments are moved to this code, and the number 0 will be displayed next to the first selected code (“Code A”), i.e., there are no coded segments under this code anymore.
3. Now “Code A” can be deleted, and the code name of the target code can be changed if necessary.

Create a New Top-Level Category and Assign Existing Codes as Subcodes

New codes are always added to the “Code System” in the selected row, i.e., where the blue focus bar is located. If you want to create a new code at the top level, first click on the root of the “Code System,” i.e., the row labeled “Code System.” Next, click on the *New code* icon or select this option from the context menu. You can then enter the name of your new code, and, if desired, assign a color. You can also assign other existing codes as subcodes of this new code by dragging and dropping them onto the new code with your mouse.

Organizing Your “Code System”

The “Code System” can be sorted automatically or manually. The automatic settings sort the codes either alphabetically (in ascending or descending order) or by code frequency (in ascending or descending order). Manual sorting allows you to determine the sequence entirely as you wish. We recommend that you start sorting the codes from the top level downward. In case of a code system with many codes, it is a good idea to hide all the subcodes first. You can then move your codes back and forth with the mouse and arrange them as required. To do this, simply move the code to the relevant position while holding down the mouse button and drop it there. Note: if a code is dropped directly onto another code, it will become a subcode of that code.

Creative Coding: A Tool for the Visual Development of a Coding Frame

“Creative Coding” is an innovative visual tool for building a structured category system. Open coding can lead to a large number of codes that are difficult to manage and difficult to organize in the “Code System.” “Creative Coding” supports the creation of a meaningful structure: you can sort and organize codes, define relationships between them, insert parent codes, and form a hierarchical structure of codes.

On a screen with a lot of space—at least if you use MAXQDA on a modern desktop computer—you can move the codes around and group them in a meaningful way. Codes that belong together in terms of content are placed close to each other, additional codes can be inserted, and codes can be renamed and assigned a color. In this way, you can create a suitable coding frame in a step-by-step manner.

“Creative Coding” involves three phases:

1. First, drag all the codes you want to organize from the code system onto the workspace. You can create a first, provisional order by manually placing thematically similar codes close to each other.
2. In the second phase, you can then thoroughly sort and group the codes, and, if necessary, create new top-level codes or subcodes. You can merge codes together or turn codes into the subcodes of others. Finally, you can also assign colors to codes or groups of codes.
3. In the third phase, the changes you have made are then transferred back and implemented in the existing code system. The complete coding frame you have generated can be exported as an image file, which you can later use for presentations (e.g., on a poster for a conference) and documentation.

In the following example, “Creative Coding” is used to set up a category system on the topic, “What do people personally think is important in life?”

Step #1: Start “Creative Coding” and Select Your Codes

- After you have started “Creative Coding” via *Codes > Creative Coding*, your code system will be displayed in the left window. In Fig. 8.4 this is a long list of codes inductively formed in the course of analyzing responses to the question “What do people personally think is important in life?”
- Drag and drop all the codes you want to sort into the workspace; if necessary, you can also remove them from the map by clicking on the corresponding icon.
- Once you have selected all the necessary codes, click the *Start Organizing Codes* icon in the upper left corner of the window to start the organization phase. The normal use of MAXQDA will then be paused until you stop the “Creative Coding” process.

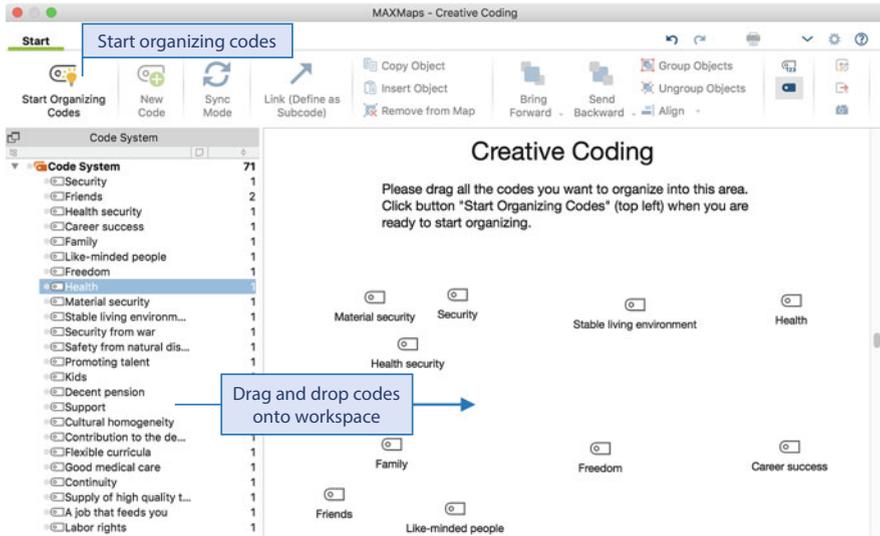


Fig. 8.4 Select codes for “Creative Coding”

Grouping and Organizing Codes

- When you start the organization phase, the code system will be hidden in the left window to maximize the workspace. At the same time, a color panel with all colors currently used in MAXQA will appear in the right margin.
- All codes can now be freely arranged and linked to each other with the mouse across the entire workspace. You can also select and move multiple codes at once by dragging a frame around them with the mouse and then repositioning the entire group.
- Relationships between codes are represented by arrows: if an arrow points to a code, it means that this code is a subcode of the other. “Code A” can be turned into a subcode of “Code B” as follows: click on the **Link (Define as Subcode)** icon to switch to Link Mode (Fig. 8.5). Then click on your desired top-level code and drag an arrow to the desired subcode while holding down the mouse button. To avoid circular relationships, any existing assignments of the subcode to other codes are automatically removed.
- By clicking the **Link (Define as Subcode)** icon again or by simply clicking on the workspace itself, you can return to the regular Selection Mode at any time.



Fig. 8.5 Important icons in the “Start” ribbon tab in Creative Coding

- It is often necessary or useful to combine several codes under a more abstract term. For this purpose, you can create new codes by clicking on the *New code* icon.
- To merge codes, be sure that Link Mode is switched off. Hold down the mouse button, and drag a code onto another code until “Merge Codes” is displayed on the mouse cursor. As soon as you release the mouse button, you will be asked if you want to merge the codes. If you click “Yes,” the code you moved will disappear from the map, and its coded segments will be assigned to the target code when you complete the “Creative Coding” process.
- The color and appearance of codes can be changed quickly and easily. To change the color, select the respective code—you can select multiple codes by dragging a frame around the codes with your mouse. You can then either select a previously used color or define a new color in the color panel on the right side. The appearance of individual codes can also be configured by clicking on the respective code icon. In the window on the right side of the screen, you can then define the font size, icon size, and more. Changes can be undone step-by-step by clicking on the corresponding icon in the upper right-hand corner of the window.

Transferring the Generated Structure of the Codes Back into the Code System

None of the actions and changes you make while working with “Creative Coding” will have any effect on your “Code System” until you click *Quit Creative Coding*. At this point you must decide whether you want the new structure you have created to be transferred to your code system or not. If so, all the codes in the “Creative Coding” workspace will be inserted into the “Code System” according to their hierarchical position and sorted alphabetically, the changed colors of the codes will be adopted, and merged codes will be merged. At the same time, the “Creative Coding” workspace will be inserted into MAXMaps as a new map (see Chap. 17) and added to your list of maps for this project—this is very useful for documenting the development process of your coding frame.

Figure 8.6 shows the result of the grouping process: eight main categories were defined on the topic of “What is important in life?” including “Personal well-being,” “Primary network,” and “Secondary network.” The codes created in the open coding phase, on which the main categories are based, are each linked to them by arrows. These do not necessarily have the function of subcategories for further analysis and further coding. In most cases they tend simply to be examples, and only during the next step are subcategories systematically formed for the eight main categories. Before doing so, however, it may make sense to code some more texts in order to test whether these eight categories can actually capture everything that is personally important in life for the research participants. Of course, new codes can also be created and arranged in a further iteration of the “Creative Coding” process during the course of the analysis.

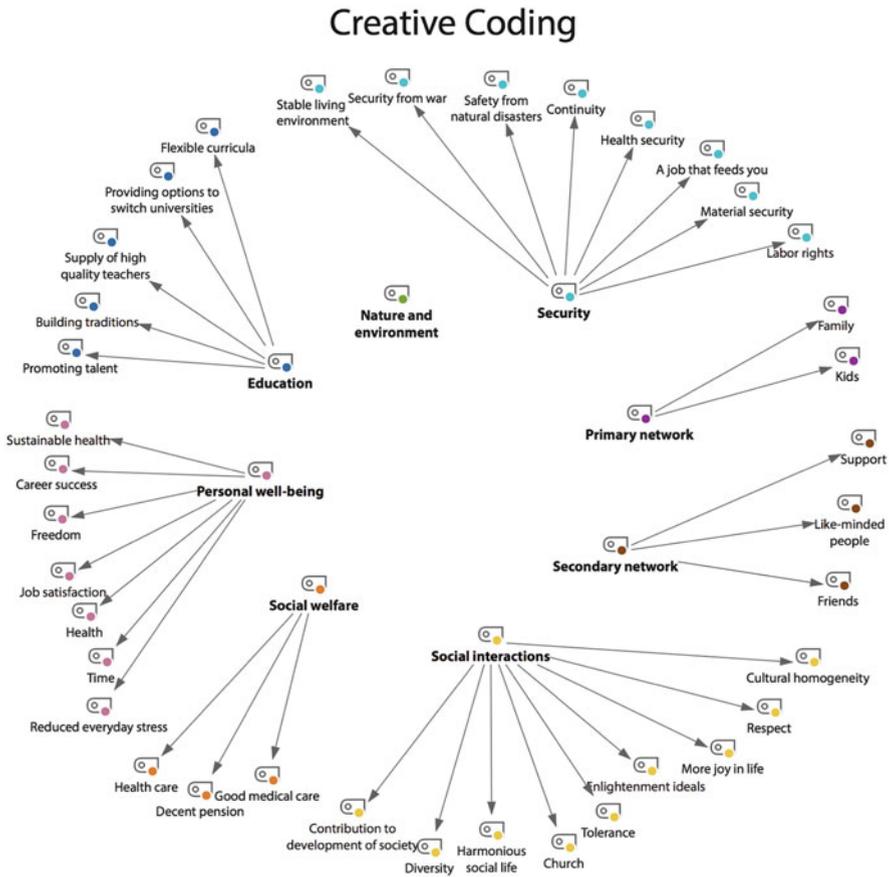


Fig. 8.6 “Creative Coding” at the end of the grouping process

Documenting the Evolution of the Coding Frame

In data-driven category formation, the category system develops in a step-by-step manner. It makes sense to document this development process and, for example, to record why categories were merged or why different categories were aligned with regard to their level of abstraction. This documentation helps to keep an overview of the development process and to be able to report on it in a comprehensible way later if required. This is also an important step as an audit trail in which the sequence of all actions in the analysis phase is recorded. An audit trail of this kind is an important quality criterion (not only) for qualitative research, as it makes the acquisition of new insights as well as the decision-making process during the analysis transparent.

MAXQDA offers various options for such documentation: firstly, an ongoing research diary can be kept in the logbook (to be found under *Home > Logbook*);

secondly, all relevant decisions can be recorded in free memos; and, thirdly, the status of the coding frame, as it is displayed in “Creative Coding” at certain points in time, can be saved as a “standard map” in MAXMaps.

Tips for Coding Frames

Finally, here are a few general hints and tips for designing coding frames:

- Do not define too many codes and keep the number of levels manageable. Codes are tools for analysis; in a toolbox with hundreds of tools, you wouldn’t find some things when you need them.
- The number of categories should normally not exceed 20 top-level categories, and the number of subcategories per top-level category should not exceed 10.
- Usually, each code name should only exist once in the code system, because MAXQDA offers many possibilities to search for the common occurrences. Duplicate code names can therefore lead to unintended results. Only in exceptional cases should the same subcodes occur under different main categories, for example, in attitudes and behavioral research if attitudes and behavior are coded in different areas. Here, it makes sense to define the behavior areas as subcodes of the main categories “Attitudes” and “Behavior,” for example, “Attitudes > Mobility” and “Behavior > Mobility.”
- Always remember that the coding frame has the purpose of organizing the data and systematizing it with regard to the research question. It is not a rigid scheme; you should be open to changes and avoid seeing the code system as possessing the character of a code of law.

References

- Charmaz, K. (2006). *Constructing grounded theory*. Thousand Oaks, CA: SAGE.
- Glaser, B. G., & Strauss, A. L. (2009). *The discovery of grounded theory: Strategies for qualitative research* (4th ed.). New Brunswick: Aldine.
- Hopf, C. (2016). In W. Hopf & U. Kuckartz (Eds.), *Schriften zu Methodologie und Methoden qualitativer Sozialforschung*. Wiesbaden: Springer VS.
- Kuckartz, U. (2014). *Qualitative text analysis: A guide to methods, practice & using software*. Thousand Oaks, CA: SAGE.
- Kuckartz, U. (2016). *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung* (3rd ed.). Weinheim: Beltz Juventa.
- Mayring, P. (2014). *Qualitative content analysis: Theoretical foundation, basic procedures and software solution*. Klagenfurt. Retrieved from <http://nbn-resolving.de/urn:nbn:de:0168-ssoar-395173>
- Schreier, M. (2012). *Qualitative content analysis in practice*. Thousand Oaks, CA: SAGE.
- Strauss, A. L., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: Sage Publications.